Visualization and Data Analysis Challenges and Solutions at Extreme Scales

James Ahrens, Jonathan Woodring, Sean Williams, Christopher Brislawn, Susan Mniszewski, Patricia Fasel, John Patchett, Li-Ta Lo
Los Alamos National Laboratory, Los Alamos, NM 87544
E-mail: ahrens@lanl.gov

Abstract. Over the past decade, the Department of Energy has led the world in enabling large-scale scientific computing as an indispensable tool for scientific discovery in a broad range of disciplines. As the scale of scientific computation has exploded, the data produced by these simulations has significantly increased in size and complexity. Scientific users are struggling with the daunting task of analyzing their massive datasets. Therefore we need to fundamentally change how we analyze data; moving from a primarily postprocessing approach to a coordinated in situ and postprocessing solution. New data triage and reduction techniques that quantify reduction-induced uncertainties are required. In this paper we present solutions to these challenges applied to real-world problems in climate and cosmology simulations.

1. Introduction

Massively parallel systems [1] have been a great success for the scientific analysis of large-scale data, but we are crossing a threshold that is changing the nature of large-scale data analysis and visualization. The data generation rates in leadership supercomputing are several orders of magnitude greater than the data movement rates for transmission and storage [2]; in other words, simulations create data faster than what can be consumed by storage and other network-connected systems. The gap between generation and consumption rates widens at exascale supercomputing, and it will continue to widen into the foreseeable future because of the mismatch between growths in the respective technologies. This rate difference amounts to a bandwidth-limited channel for simulation data transmission and movement, which currently is the largest bottleneck for scientific postanalysis and interactive visualization. The bottleneck greatly increases the time for data movement from a simulation to the point that only a small fraction of the total generated data can be postanalyzed because of transmission time. This implies that to continue using simulation-based science, we must consider how to move and analyze large-scale data sets in a bandwidth efficient manner, so that a scientist can create discoveries from the data in a reasonable amount of time.

2. In Situ Science-Based Feature Extraction

Most if not all science depends on “features” or abstractions for scientific analysis and discovery. Rather than looking directly through raw data to understand complex scientific phenenoma, we generally simplify and categorize complex details for our “constant” cognitive bandwidth [3], to understand, explain, and communicate. This is doubly true for large-scale, complex simulations. While in the past scientists were able to look at the data points individually to make sense of them, the current scales of data make this task impossible. Therefore, a key component for current and future large-scale visualization and analysis is the integration of science-based feature detection and extraction methods, to reduce the massive size of our data to an understandable level. Also, integrating science oriented abstractions directly into the simulations improves the performance of simulations by reducing the amount of time spent in simulation I/O, since features and extracts take less storage space, while also amortizing postprocessing analysis time for faster workflow and analysis.

2.1 Large-Scale Cosmological Simulations and Halo Finding

For large-scale cosmological simulations, such as the Roadrunner Universe MC^3 [4]—an n-body particle simulation for dark matter—many of the interesting scientific questions are based on cosmological structure. One particular feature of interest is the clustering of dark matter particles in a simulation, known as "halos." The cosmological community has several different definitions for halos, including friends-of-friends (FOF) and spherical overdensity (SO). During the creation of the new dark matter simulation code, MC^3, computer scientists were brought in at the start of the project to provide visualization and computing expertise. This approach has allowed us to create sophisticated science-based feature extraction methods, including a fast halo finder available as a simulation-time analysis code and as a postprocessing method in a large-scale visualization tool.

The FOF halo-finding algorithm finds all clusters of particles that is similar to finding 3D connected components. A particle is considered a friend of another particle if they are within threshold distance or "linking length." A halo is a complete set of friends-of-friends, or a set of particles where all particles are within the linking length of at least one other particle in the set. The original implementation used by the cosmologists would take hours on a relatively small data set. Because of the slow speed, this made it unsuitable for in situ analysis (running visualization and analysis during a simulation run) and interactive postanalysis and visualization. Working with the cosmologists, we were able to accelerate the FOF halo-finding algorithm with spatial search data structures and parallelization of the algorithm [5],[6]. With 256 processors, we are able to locate all of the FOF halos in 20.5 s for a billion particles (see Table 1), opposed to previously taking hours for the same task.

Table 1: Strong scaling results for our fast, parallel FOF halo-finding algorithm for 1 billion particles. The original implementation would take hours, while the algorithm now takes seconds or minutes, making it suitable for in situ analysis and interactive visualization. We also have good weak-scaling results, due to little to no communication between the processors during halo finding.

Number of Processors	Completion Time for FOF Halo Finding for 10^9 Particles
64	66.6 seconds
128	32.9 seconds
256	20.5 seconds

Halo finding is an attractive algorithm for in situ feature extraction because of the speed, amortization of analysis costs, and output data reduction that occurs at simulation time. For example, if only the halo particles are kept, we can get at least a 50% reduction in storage size, providing a 50% speedup of postanalysis read and simulation write times. If we consider one of the initial and largest MC^3 runs with 64 billion particles, the cosmologists were able to save much more space and time by saving only the particle centers for the halos. The halos are approximately thousands of particles in size; therefore, saving halo centers represents several orders of magnitude in space and time savings compared with raw particle data. More recently, we have also integrated sub-halo finding and spherical overdensity (SO) [6], adding to the growing suite cosmological analysis algorithms, which can save storage space, simulation time, and postanalysis time through in situ analysis.

Our halo-finding algorithm is also provided as a postanalysis and visualization filter to match the in situ implementation, which is integrated into ParaView [7], a large-scale, parallel visualization tool built on top of the Visualization ToolKit [8]. ParaView and VisIt [9], end-user large-scale visualization tools funded by Department of Energy, are among few visualization systems that are able to analyze the large-scale data generated by DOE scientists and academic collaborators. With the halo finder in a visualization tool, we have shown the utility of qualitative and quantitative analysis, combined with provenance, for scientific discovery [5]. The cosmologists were able to combine visualization and analysis to perform a cosmological code comparison task of several different simulation codes. The workflow used by the cosmologists is a step-by-step process: a domain scientist formulates a hypothesis, then generates insight with visualization, verifies insights with quantitative scientific analysis, and makes conclusions using the collected scientific provenance during the process. This process has allowed the cosmologists to create new scientific discoveries, because of the interactive feedback loop through combining visualization and quantitative analysis that connects scientist insight to evidence.

2.2. Eddie Finding

We now present a detailed example of the utility of feature detection technique, both model validation and data reduction, applied to ocean climate science.

Ocean eddies are important in transporting water properties. These affect the location and strength of ice sheet melting, phytoplankton blooms, and ultimately heat and carbon uptake from the atmosphere to the ocean. The details of the ocean dynamics change the rate and distribution of atmospheric warming, and so the study and verification of eddies in ocean simulations are important in climate change analysis. Some example eddies are shown in Figure 1. At any given time, one may identify thousands of individual eddies in the sea surface height field, as recorded by satellite altimetry studies [10]. These eddies are spawned in energetic currents, such as the Gulf Stream, Kuroshio Current, and Antarctic Circumpolar Current, and can then live for months to years and propagate thousands of kilometers. During its lifetime, an eddy can isolate waters from its birth region and transport the water properties across ocean basins, altering the temperature, salinity, and biological productivity along this route. Thus the study of eddies and the quantification of their effects is important for understanding water mass properties, ocean dynamics, and nutrient transport.

The role of eddies can be investigated in two ways: through observations and model simulations. Satellite data is available for the whole globe from 1992 on, at relatively high resolution, but provides only surface information. Measurements from ship surveys and autonomous floats provide data below the surface but are sparse and follow only single tracks. On the other side, simulations are an effective tool to study ocean dynamics, and can provide full three-dimensional ocean data at high resolution in time and space. Scientists at Los Alamos National Laboratory develop ocean models for high-performance computing platforms that are coupled with atmospheric and sea-ice components to study the effects of climate change. Thus, ocean simulations fill a special need in climate research, but they must be verified through direct comparisons with observations whenever possible. Mean quantities such as surface velocities and volume and heat transport are common variables for comparison.

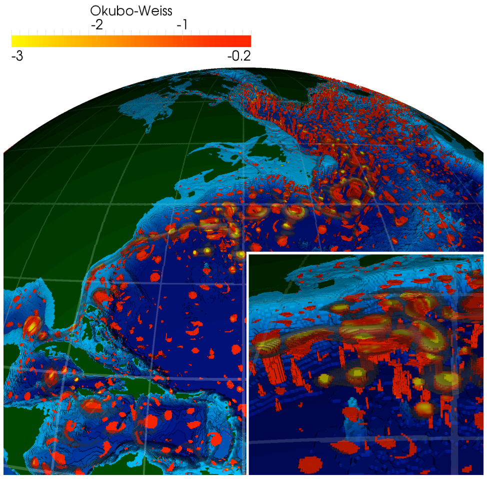

Figure 1: The Q-criterion, a standard metric in oceanography for extracting two-dimensional vortices, visualized at depth by thresholding all low-vorticity points. The three-dimensional shapes of the eddies are made clear: in this region containing the Gulf Stream, several strong eddies reach deeply into the ocean, while smaller eddies remain near the surface, and the Gulf Stream itself dominates only near the surface.

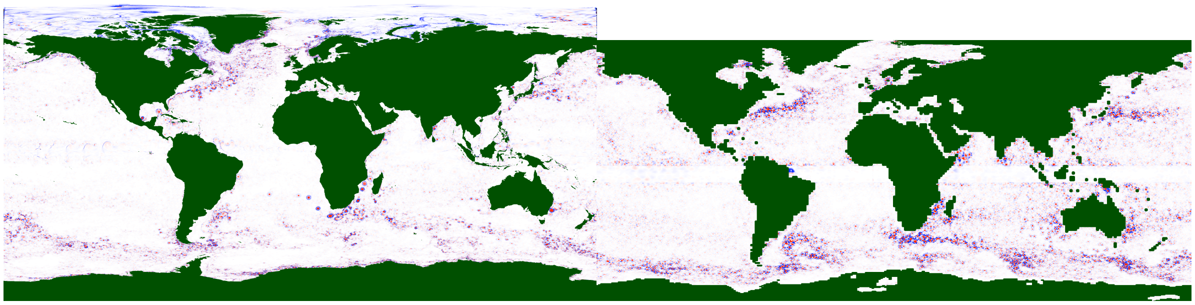

Figure 2: The Q-criterion, visualized globally at the surface for POP simulation data (left) and Aviso satellite data (right). Points colored red have high vorticity, and the more circular red features are generally eddies.

To compare eddies between POP simulations and observational satellite data, we must assess the validity of our definition of an eddy against theoretical expectations of eddy behavior. We define an eddy based on the Q-criterion [11], using thresholds selected based on several quality metrics. The criterion itself is visualized directly at the ocean surface in Figure 2, while an example thresholding is shown at depth in Figure 1.

Features such the Q-criterion are useful in reducing the amount of data that needs to be saved from the simulation. A typical run of the Parallel Ocean Program (POP) saves floating-point values in a 3600 x 2400 x 42 rectilinear grid of single-precision floating-point values. To identify vortices, we require, at minimum, the zonal (east-west) and azimuthal (north-south) components of velocity, resulting in a total size of 2.8 GB per time step. Our vortex identification method identifies about 5000 vortices in a typical time step, and in order to do further processing (for example, tracking) of the vortices, we require at minimum two position values (longitude and latitude), two velocity values (zonal and azimuthal), and radius. To do vortex extraction in situ would require writing out about 25,000 floating-point values per time step, or about 100 KB, or about 0.003% of the total data.

Figure 2 shows qualitative agreement between simulation and observation; for example, most significant eddy activity occurs in the Gulf Stream (off the east cost of the United States), the Kuroshio Current (off the east coast of Japan), and the Antarctic Circumpolar Current (ringing Antarctica, south of Africa and the Americas).

We now look at comparing the simulation and observation more quantitatively. Statistics from an eddy census of ocean simulations provides more detailed analysis of the ocean dynamics for comparison. Just like the weather, the size, location, and path of an individual eddy are due to chaotic effects and cannot be directly matched between models and observations. However, average statistics from thousands of eddies should match, and demonstrating this agreement provides confidence that model studies faithfully represent the real ocean. For example, the average radius of an eddy should be smaller at the poles and larger at the equator, because the horizontal Coriolis force (caused by the rotation of the earth) is strong at the poles and weak towards the equator. This has been observed in statistics accumulated over 16 years of satellite observations [10].

To make a quantitative comparison, we need to register the simulation and observational data. The satellite data is at a resolution of 1/3 degree, while the simulation data is at a resolution of 1/10 degree. To compensate, we consider eddies in the POP data only if they have a radius of at least three grid cells, and the POP eddies have their radii discretized to 1/3 degree. Beyond that, the POP data is on a tripole grid to correct for the singularity induced at the poles by dividing a sphere into a regular grid. The Aviso satellite data is derived from altimetry (sea surface height), and velocity is derived from the fact that, due to the near-incompressibility of the ocean, contours of sea surface height are also streamlines of velocity. While the satellites are circumpolar, little data is collected in the Arctic Ocean (particularly because of the sea ice covering most of it), and data near the equator is unreliable or nonexistent (because of the extremely weak Coriolis force).

Eddies are tracked by assuming the entire eddy is translating in the direction and speed of the average velocity of all points in the eddy. That is, imagine a spinning hockey puck sliding along a rink, and consider the velocities of each point on the puck: any vortical component of the velocity will be canceled by an opposite vortical component on the opposite side of the puck, so only translational velocity will remain. We then use this information to build a tracking graph, by considering two eddies in adjacent time steps to be the same if an eddy, offset by its velocity (in degrees per day) times one day (the outputted temporal resolution of the simulation), overlaps another (non-offset) eddy in the next time step.

Because of the relationship between latitude and Coriolis strength, we expect eddies to be small at the poles, and to get larger approaching the equator. However, making a plot of the average radius of all eddies per degree of latitude, in Figure 3a, we see (in the red line) that eddies get larger to about -50 degrees from the south and about 40 degrees from the north, then flatten out. We believe this behavior is caused by eddies becoming weaker near the equator and showing a correspondingly weaker signal in the Q-criterion. These weaker eddies appear smaller when thresholded, so they reduce the average radius near the equator. We observe, however, that if we use the tracking information to consider only long-lived eddies (we use a three-month lifetime), these weaker eddies are thrown out, and the resulting plot of latitude versus average radius (the blue line) matches theoretical expectations quite well.

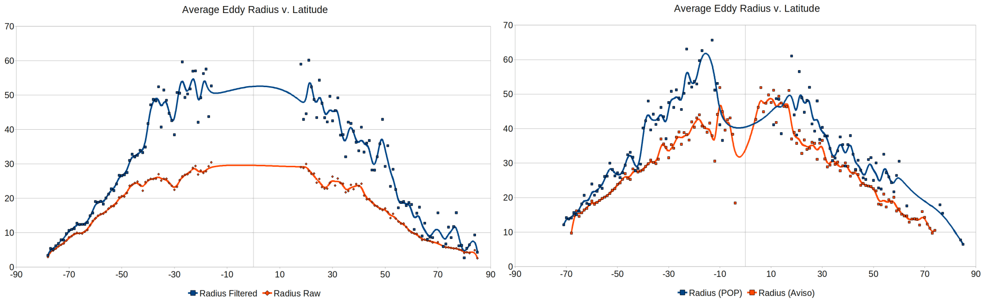

Figure 3: (a): Average eddy radius as a function of latitude. Because of the Coriolis force, eddies get larger as they get closer to the equator. However, direct application of the Q-criterion (red line) results in the average eddy radius flattening out well before reaching the equator. By considering only eddies that live for at least three months (blue line), the correct distribution shape is achieved. (b): Average eddy radius as a function of latitude for both the POP simulation data and Aviso satellite data. For this comparison, the radii of POP eddies were discretized to 1/3 degree (the resolution of the Aviso data) in order to make their sizes directly comparable. We conclude from this plot (in addition to the above surface maps) that the POP simulation is producing similar eddies to those observed in real data.

We also compare the POP data with observational data gathered from satellites. For this comparison, we consider an eddy only if it lives for at least three months. This sort of feature-based comparison is especially useful because of the many structural differences between the two data sets. To make the comparison, we generate a plot of average radius versus latitude for the two sources (Figure 3b). Our results show that the curves for simulation and satellite data have good correspondence.

3. Quantifiable Data Reduction for Postanalysis

As a data reduction method, in situ visualization and analysis [12], [13] greatly reduce the amount of simulation I/O time by several orders of magnitude by writing out predefined visualization and analyses: extracts, features, images, movies, and so forth, instead of full-resolution, raw data sets. The main drawback with this method is that it is difficult to perform exploratory analysis because in many cases the operations are predefined and most of the parameters are fixed, leaving the exploration to a few free parameters, such as view direction for isosurface extracts or the color map for explorable volume rendering images [14]. To allow for a much wider range of options in exploratory post-analysis, we look to continue to save some raw portion of large-scale data through data reduction. But, it is not enough to reduce only the data size; we also need to measure the precise effect of information loss on scientific analysis. Information loss can come from a variety of sources: dropped data points, loss of precision, nonuniform data transformations, and so forth, and it can have different effects on results depending on the specific postanalysis operations and the scientific domain. Therefore, we are researching methods to use data reduction on large-scale data, while also quantifying the effects on visualization and analysis operations and scientific results.

The bandwidth bottleneck can impede the scientific workflow at different points in the postanalysis pipeline: when data is written to storage, when data is moved from a remote site, and when data is read from storage. We describe three of our research solutions to cope with these data movement bottlenecks. First, to cope with writing large-scale data, we have created a stratified random sampling solution for simulation particle data that scales down stored data by creating smaller data samples [15]. Second, in the movement of data from a remote site, we explore the use of compression techniques on ocean climate data to reduce the time to copy scientific data. Third, to improve read times during interactive visualization and analysis, we have integrated multiscale, level-of-detail methods into our existing large-scale parallel visualization and analysis tools [16]. In the reminder of this section, we describe each of our solutions in detail.

3.1 Data Sampling for Simulation I/O Reduction

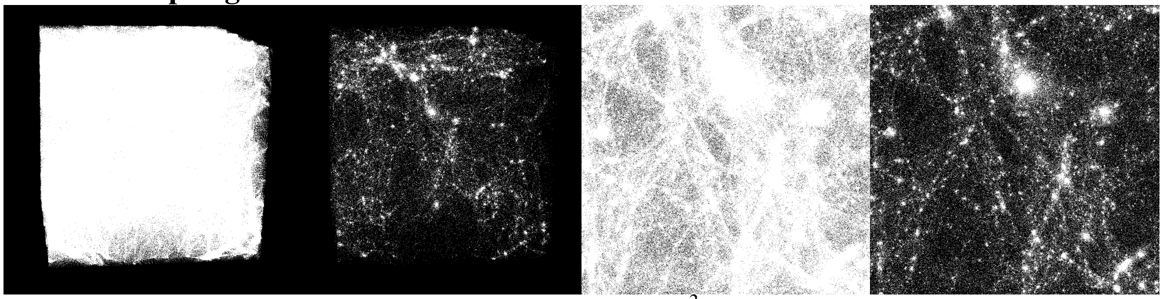

Figure 4: The difference between a full-resolution MC^3 dark matter cosmological simulation (particle data) and a 0.19% stratified random sample of the same data.

In situ sampling of simulation data is not a completely new idea: all large-scale simulations currently store a subset of the generated data, since it is not possible to store all the data [2]. One way this shows up in practice is by storing very few time steps. An explicit decision is made for temporal sampling, and the majority of information loss is pushed into the temporal domain. Furthermore, typically only a few variables (fields) are stored for postanalysis because of bandwidth and storage. We use a spatial random sampling technique on large-scale particle data (see Figure 4), in addition to temporal sampling and dropping fields. Combined with temporal and variable sampling, we allow a simulation to trade I/O time on time, space, and variable fidelity for postanalysis, spreading the information loss across multiple dimensions, rather than just time and variable loss.

Table 2: Simulation I/O time savings by applying an in situ stratified random sampling to MC^3 data. The baseline time is the time for an MC^3 2 billion particle simulation to write one full resolution time step across 512 processors to a Panasas file system from Roadrunner without level of detail (LOD) storage. At 25% sample size or less, we get a significant time and data size savings per time step, such that we can write two time steps in the same amount of time as one full resolution time step. At 1.6% sample size, we can write 12 time steps in the same amount of time that it takes to write one full-resolution time step. Time steps that have full resolution or 50% sample size data take longer to write, because of the time and space overhead of processing the data into an LOD structure, but there still is an end-to-end time savings in the overall analysis workflow. By processing the data in situ for LOD storage, we save 364 s per time slice, by removing one read (153 s) and one write (210 s) of the data, before visualization and analysis.

Data Sample Size	Write Time	Total Data Size Ratio	Write Time Difference	Write Time Ratio
Baseline (full resolution data without LOD)	210.5 s	1.0	0.0 s	1.0
Full resolution with LOD	509.2 s	2.0	-298.7 s	0.41
50% sample LOD	261.6 s	1.0	-51.2 s	0.80
25% sample LOD	92.3 s	0.33	118.2 s	2.28
12.5% sample LOD	41.3 s	0.14	169.1 s	5.09
1.6% sample LOD	17.5 s	0.016	193.0 s	12.02

The sampling methodology that we used, for the Roadrunner Universe MC^3 simulation, is a stratified random sampling [17] of the particles across space. This is implemented by using the spatial decomposition of particles across the simulation processors and partitioning the particles within a processor via kd-tree decomposition. We chose this approach for three reasons: (1) stratified sampling generates a better sample than simple random sampling by evenly spreading the samples across space, (2) the kd-tree decomposition naturally maps our sampling algorithm into a level-of-detail, hierarchical storage for interactive visualization [16], and (3) we achieve significant space, simulation I/O time, and postprocessing I/O time through in situ sampling and directly writing the data into a level-of-detail format from the simulation (see Table 2) [15].

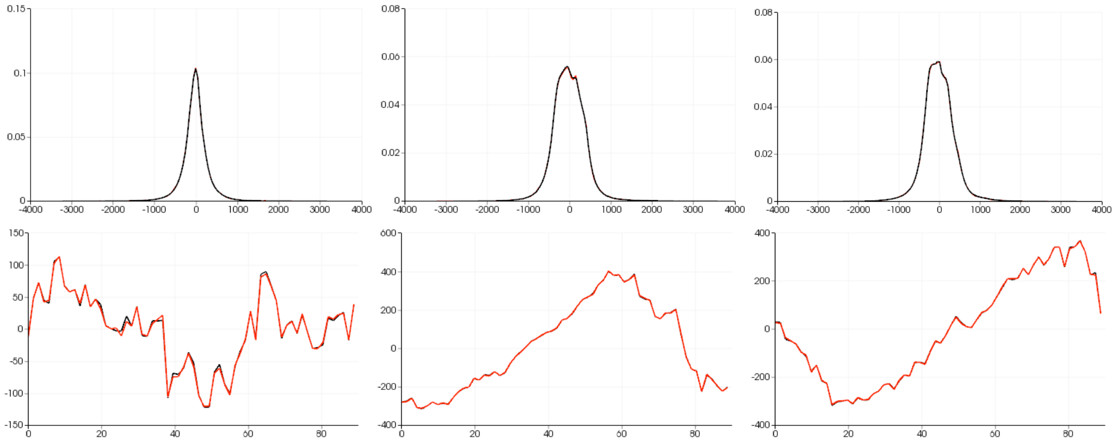

Figure 5: Comparison between 0.19% sample data (red) and raw MC^3 data (black). The top graphs show both data sets as histograms in the velocity components (black occludes red). The bottom graphs show both data sets in the velocity component value as it varies across space (red occludes black).

In our studies, we show that MC^3 sample data is able to accurately replicate various statistical measures (see Figure 5) [15], such as histograms, and scalar value distributions across space, when compared with the full-resolution data, down to a 0.19% stratified random sample of the original data. This reduces the I/O size per time step by three orders of magnitude, yet it is able to accurately replicate the original data in these measures. In a more complex analysis we found that with halo finding (clusters of simulation particles), we were able to replicate the histogram of halo masses (an important cosmological measure) at a 1.6% sample size (see Figure 6), representing a savings of two orders of magnitude in storage and simulation I/O time. This shows that we were able to significantly reduce the amount of necessary storage space and I/O time, while achieving relevant scientific results.

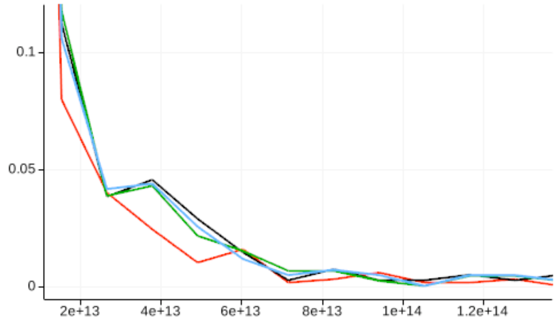

Figure 6: Comparison of halo mass histograms at different sample sizes. Black is the original full-resolution data, while red is a 0.19% sample, green is a 1.6% sample, and blue is a 12.5% sample.

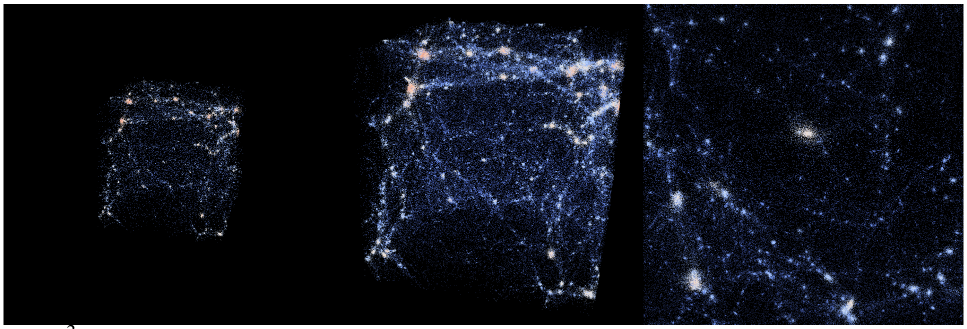

Figure 7: MC^3 particle sample data augmented with statistics of the original data set. The points are colored by the local variance of the original data, which highlights portions in the sample data that are uncertain compared with the full-resolution original data.

We also quantify the error and uncertainty [18] in sample data, compared with the original data, by augmenting it with localized, aggregate measures of the original data [15]. As we generate a random sample of the data during the simulation, we calculate properties of each sampling stratum, such as statistical properties. These measures, such as statistical variance, allow us to compactly describe the local uncertainty around a sample point. For example, in visualization of sample data, we are able to highlight sample particles that have a wide variance in the sample strata for a variable of interest (see Figure 7). This is a key means for users to show sample uncertainty due to information loss. Techniques, such as variance highlighting shows that there potentially is a complex feature in higher-resolution data for a local sampling stratum, which was not captured at a particular sample size.

3.2 Data Compression for Remote Data

At Los Alamos National Laboratory (LANL), the climate scientists run daily Parallel Ocean Program (POP) simulations [19] on Oak Ridge National Laboratory (ORNL) supercomputers. In addition to using the remote supercomputer for distance visualization via images and for batch movie generation, there is a need to move the data to LANL for local analysis by collaborators and team members. Because of the large size of the data (1.4 GB per variable per time step) and the low single-link bandwidth between LANL and ORNL (measured to be under 1 MB/s for a serial link), it takes approximately 23 minutes to copy one variable for one time step. Since typically climate scientists are interested in four variables (salinity, temperature, east-west velocity, north-south velocity), the total is 5.6 GB per time step, and a transfer time of 72 minutes per time step. A normal-sized set of time steps can have from tens to thousands of time steps, bringing the total data size up to 56 GB for ten time steps or 5.6 TB for a thousand time steps. The transfer time in this case would take 12 hours for the former and 50 days for the latter case.

We explore using compression techniques to reduce the transfer time and required bandwidth for large-scale data movement. One area of interest is wavelet compression and decomposition, because of the multiscale nature of the method. We implement our wavelet compression by using the state-of-the-art JPEG 2000 technology [20],[21] from the signal processing and compression community, leveraging their knowledge and allowing us to reduce the bandwidth requirements for large-scale scientific data. Assuming we have raw floating-point data generated by the POP simulation, we perform a series of steps to compress, transfer, and decompress the data from LANL to ORNL.

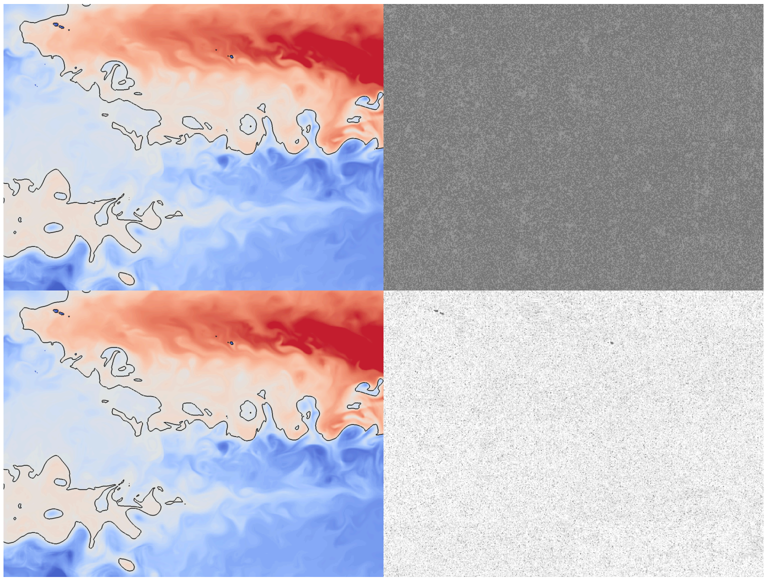

Figure 8: POP ocean climate data that is wavelet compressed (JPEG 2000) at 8 bits per pixel (top) and 2 bits per pixel (bottom). The standard metric to assess the quality is average image difference, and in the visualization of the data (left images), it is hard to see the difference between the two. If we compare the pointwise data differences between compressed and uncompressed data (right side, log-scale coloring) there is a large discrepancy between the two compressed data sets, which has a direct effect on scientific analysis. The top data have a maximum error of 1.49012e-08, while the bottom data have a maximum error of 5.23031e-06, representing a loss of two digits of precision.

The primary quality assessment for compressed data in visualization has been average final image difference between compressed and uncompressed rendered data, such as root mean squared error (RMSE) and signal-to-noise ratio (SNR). While this is good for measuring average performance of the compression and visual fidelity, it is not good enough or conservative enough of a metric for scientific analysis. SNR and RMSE can “hide” high pointwise errors that can affect scientific analysis results. Those specific points can create problems for local or accumulated analysis because the error is not bounded nor described for a scientist (see Figure 8). Instead, we calculate maximum error (L-infinity norm) rather than average error (L2 norm) to provide a 100% maximum error guarantee. We do this so a domain scientist is assured on the minimum precision, because maximum error translates to minimum precision such that the data is precise to x decimal places, beyond which the scientist can expect noise with increasing probability in the least significant bits. This shifts the emphasis in compressed data from qualitative image difference in visualization to precise data error management for science-based analytical tools. Furthermore, many visualization methods, such as isosurfacing and thresholding, would be qualitatively and quantitatively improved through precise error bounds.

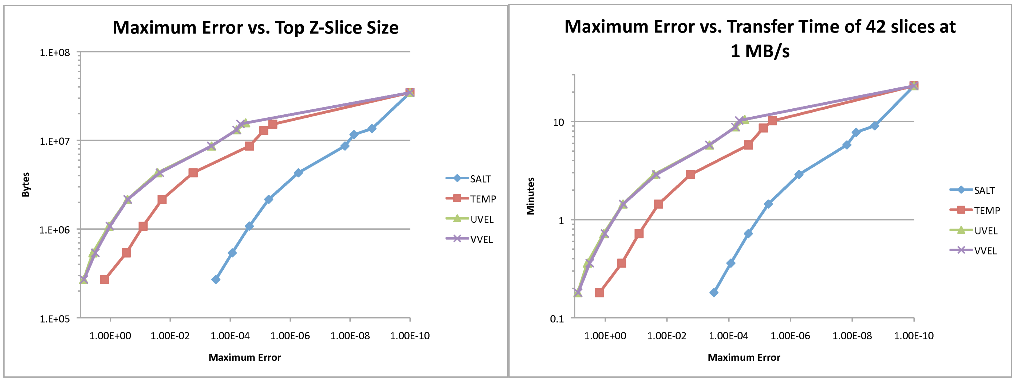

Figure 9: Charts that show the tradeoff between maximum error and data size or transfer time for POP compressed data. The graphs allow the climate scientists to make an informed choice of the time taken to transfer the data at a particular maximum error, allowing them to trade between time and data quality.

Measuring maximum error allows our climate scientists to make an informed choice between data precision and transfer time over a bandwidth-limited link. We provide error graphs that show the relationship between maximum error and the data size and transfer time (see Figure 9), allowing a domain scientist to trade between data accuracy and transfer time in a controlled and quantified way. The upper-rightmost point in Figure 9 is the data size and time for a single-field uncompressed POP data, which takes 23 minutes to transfer in our LANL to ORNL 1 MB/s case. Though the uncompressed data is "infinitely" precise, we put the point at 10^{-10} to be able to graph it. The points down and to the left are the results from JPEG2000 compression at various bitrates, with increasingly faster transfer times and larger, but quantified, maximum errors.

3.3 Multiresolution, Level-of-Detail Remote Visualization

The final hurdle for large-scale scientific analysis is interactively delivering the data to the scientist while coping with the read and display bottleneck (there are many orders of magnitude between the size of the data generated and what can be seen). At the current levels of supercomputing, nearly all large-scale data visualization and analysis scenarios are considered to be at a distance, because of the storage, I/O, and network bandwidth limitations [2]. Standard practice for distance visualization is through the delivery of images, but the interactivity is limited to the network bandwidth for frames per second and network latency for response time. This method of distance visualization also requires a large parallel visualization system running on the remote site, such as a visualization cluster or a supercomputer. Additionally, massively parallel distance visualization do not address the display bottleneck, since all the data is rendered to a much smaller display or image, introducing an implicit, regular sampling of the data.

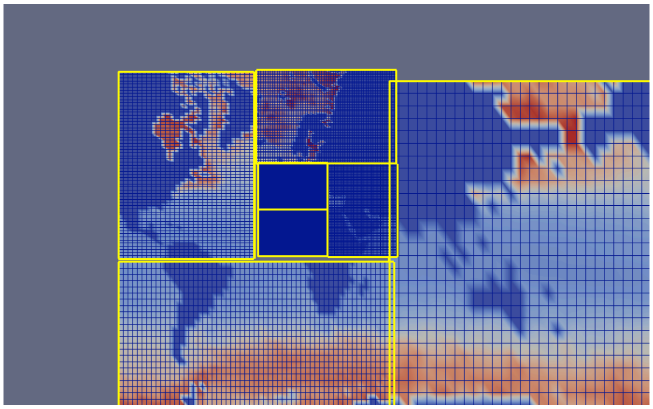

Figure 10: Out-of-core, level-of-detail visualization of POP data in ParaView. We leverage our existing large-scale visualization framework and extend it with a streaming, multi-resolution approach to account for the largest bottleneck for interactive visualization: storage, I/O, and display bandwidth.

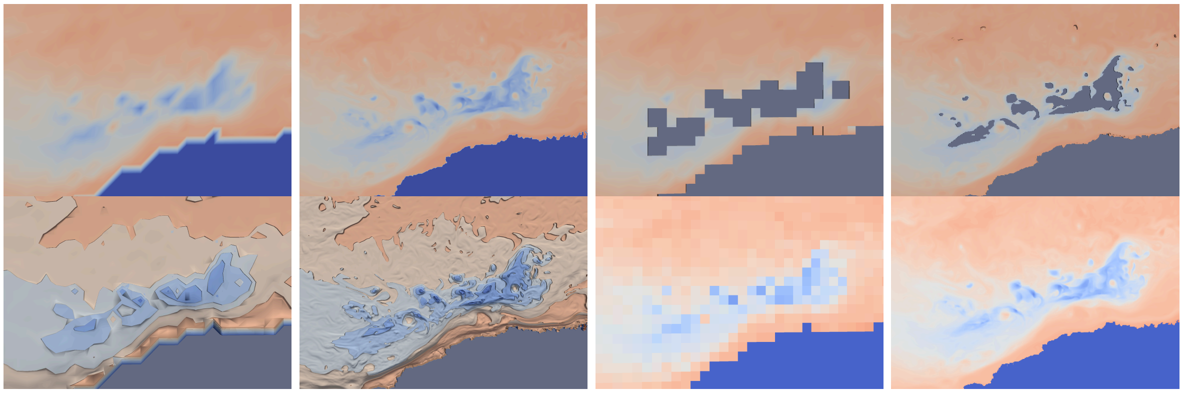

Figure 11: Existing visualization operations applied in a streaming, multiresolution approach. From left to right, top to bottom (in groups of two – low-resolution and high-resolution for each operator): surface rendering, thresholding, isosurfacing, and point rendering.

As an alternative to image-based distance visualization, we utilize a representative-based (data or object) distance visualization using out-of-core (streaming) and level-of-detail (multi-resolution) techniques (see Figure 10) [22],[16]. Representative-based visualization and analysis sends raw data over the network, rather than images, by interacting with the remote storage system rather than a remote computing resource. This allows us to decouple the visualization and analysis task from the remote supercomputer, using local computing resources rather than supercomputing time. This distance visualization method also allows for higher interactive frame rates that are not limited by the network bandwidth and latency by using local rendering resources. We integrate out-of-core and level-of-detail methods into our distance visualization to progressively update the data to meet target response times. Out-of-core level-of-detail methods visualize and analyze data at various resolutions, such as using coarse, low-resolution data for contextual overviews while progressively updating to high-resolution data for zoomed-in, focus details. This approach allows large-scale data to be shrunk down to displays and low-bandwidth networks, and still provide interactive full-resolution details on demand.

We have integrated this streaming, multiresolution framework as an extension to parallel VTK and ParaView [7],[8], such that the existing large-scale visualization and analysis operators (see Figure 11) can be directly applied to the data, reusing the past research and implementation done by the visualization community. We also combine and leverage our previously described research in data compression and data sampling [15]. Both the JPEG2000 compressed data and randomly sampled particle data is usable as multi-resolution input data, which provides interactive, streaming details on demand.

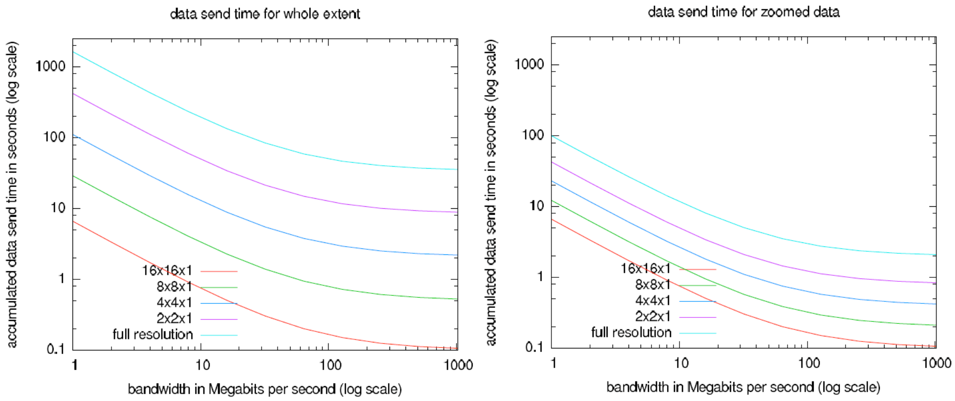

Figure 12: Data send times for streaming POP data at various resolutions in a representation-based multiresolution version of ParaView. The different curves represent the time to completion at various resolutions ranging from full-resolution to 16x16 regular sampling in x and y. Network bandwidth with 100 ms latency varies on the x-axis and the resulting time to completion results on the y-axis. The graph on the left represents the whole spatial extent, while the right graph represents a zoomed-in focus area. Even at low bandwidth, the left graph shows that a user can quickly get the whole spatial extent at low-resolution. The right graph shows that for a zoomed-in portion, a user can get full-resolution data in a few seconds, getting details on demand.

The following timings of our streaming, multiresolution distance visualization were done with the POP data set, where the single-field, single time-slice data set at full resolution is 3600 × 2400 × 42 at 1.4 GB. We used regular sampling of the data for the multi-resolution representation, where we refer to specific resolutions of the data as n × n × 1 to represent the strides of the full-resolution data in x and y, with no sampling in z. Our multiresolution, streaming visualization sends a low-resolution representation at first, which is shown almost instantaneously because of the small size (see left graph of Figure 12). It will progressively stream higher-resolution data to incrementally update the low-resolution data in focus (high priority) areas [16].

At our target bandwidth for our previous Oak Ridge National Laboratory (ORNL) distance case of 1 MB/s, it takes approximately 1 second to deliver and render 16 × 16 × 1 low-resolution data. 16 × 16 × 1 is 5.4 MB of data. To put these results into context, compared with our other data reduction techniques, that would represent a 0.14 bits per pixel JPEG2000 compression or a 4.5% random sample. After the initial delivery time, the user can interact with a downscaled, overview representation interactively at high frame rates, utilizing the local rendering resources. While the user is interacting with this data, the multiresolution system streams higher resolution data to important focus areas of the visualization either automatically determined or selected by the user. It iteratively improves the quality over time, by prioritizing focus data, which saves network bandwidth and improves time to completion in the focus area. For example, a zoomed-in portion of full resolution POP data, for a particular view, is delivered in 22 seconds (right graph of Figure 12), compared with sending the whole data at full resolution, which takes at 23 minutes.

4 Conclusions

In this paper, we discussed a number of new techniques for data analysis and visualization including in situ feature extraction and quantifiable data reduction. Given the importance of understanding our massive-scale simulation outputs, these techniques and others approaches will be required to meet the data challenges of extreme-scale scientific applications.

References

[1] G.H. Weber and E.W. Bethel, with H. Childs, D. Pugmire, S. Ahern, B. Whitlock, and M. Howison, Prabhat, "Extreme Scaling of Production Visualization Software on Diverse Architectures," *Computer Graphics and Applications, IEEE*, vol. 30, 2010, pp. 22-31.

[2] J. Ahrens, B. Hendrickson, G. Long, S. Miller, R. Ross, and D. Williams, *Data Intensive Science in the Department of Energy*, Los Alamos National Laboratory, 2010.

[3] C. Johnson and R. Ross, *Visualization and Knowledge Discovery: Report from the DOE/ASCR Workshop on Visual Analysis and Data Exploration at Extreme Scale*, Department of Energy Office of Science ASCR, 2007.

[4] S. Habib, A. Pope, Z. Lukić, D. Daniel, P. Fasel, N. Desai, K. Heitmann, C.H. Hsu, L. Ankeny, G. Mark, and others, "Hybrid Petacomputing Meets Cosmology: The Roadrunner Universe Project," 2009, p. 012019.

[5] J. Ahrens, K. Heitmann, M. Petersen, J. Woodring, S. Williams, P. Fasel, C. Ahrens, C.-H. Hsu, and B. Geveci, "Verifying Scientific Simulations via Comparative and Quantitative Visualization," *IEEE Computer Graphics and Applications*, vol. 30, 2010, pp. 16-28.

[6] J. Woodring, K. Heitmann, J. Ahrens, P. Fasel, C.H. Hsu, S. Habib, and A. Pope, "Analyzing and Visualizing Cosmological Simulations with ParaView," *Arxiv preprint arXiv:1010.6128*, 2010.

[7] http://www.paraview.org/, *ParaView*.

[8] http://www.vtk.org, *Visualization Toolkit (VTK)*.

[9] https://wci.llnl.gov/codes/visit/, *VisIt*.

[10] D.B. Chelton, M.G. Schlax, R.M. Samelson, and R.A. de Szoeke, "Global Observations of Large Oceanic Eddies," *Geophysical Research Letters*, vol. 34, Aug. 2007, p. 5 PP.

[11] J.C.R. Hunt, A.A. Wray, and P. Moin, "Eddies, Streams, and Convergence Xones in Turbulent Flows," 1988, pp. 193-208.

[12] T. Tu, H. Yu, L. Ramirez-Guzman, J. Bielak, O. Ghattas, K.-L. Ma, and D.R. O'Hallaron, "From Mesh Generation to Scientific Visualization: An End-to-End Approach to Parallel Supercomputing," *Supercomputing, ACM/IEEE 2006*, Tampa, Florida: 2006, p. 91.

[13] K. Moreland, N. Fabian, P. Marion, and B. Geveci, *Visualization on Supercomputing Platform Level II ASC Milestone (3537-1B) Results from Sandia*, Sandia National Laboratory, 2010.

[14] A. Tikhonova, C.D. Correa, and K.-L. Ma, "An Exploratory Technique for Coherent Visualization of Time-Varying Volume Data," *Computer Graphics Forum*, vol. 29, 2010, pp. 783-792.

[15] J. Woodring, J. Ahrens, J. Figg, J. Wendelberger, S. Habib, and K. Heitmann, "In-situ Sampling of a Large-Scale Particle Simulation for Interactive Visualization and Analysis," *To appear in Computer Graphics Forum (EuroVis 2011)*, 2011.

[16] J.P. Ahrens, J. Woodring, D.E. DeMarle, J. Patchett, and M. Maltrud, "Interactive Remote Large-Scale Data Visualization via Prioritized Multi-Resolution Streaming," Portland, Oregon: 2009, pp. 1-10.

[17] W. Cochran, *Sampling techniques*, New York: Wiley, 1977.

[18] C.R. Johnson and A.R. Sanderson, "A Next Step: Visualizing Errors and Uncertainty," *Computer Graphics and Applications, IEEE*, vol. 23, 2003, pp. 6-10.

[19] R.D. Smith and P. Gent, *Reference Manual of the Parallel Ocean Program (POP)*, Los Alamos, NM: Los Alamos National Laboratory, 2002.

[20] M.W. Marcellin, M.J. Gormish, A. Bilgin, and M.P. Boliek, "An overview of JPEG-2000," *Data Compression Conference, 2000. Proceedings. DCC 2000*, 2000, pp. 523-541.

[21] D.S. Taubman, M.W. Marcellin, and M. Rabbani, "JPEG2000: Image Compression Fundamentals, Standards and Practice," *Journal of Electronic Imaging*, vol. 11, 2002, p. 286.

[22] E.J. Luke and C.D. Hansen, "Semotus Visum: A Flexible Remote Visualization Framework," *IEEE Visualization, 2002. VIS 2002.*, Boston, MA: 2002, pp. 61-68.